MICROCOMPUTER ARCHITECTURE FOR COMPUTER SECURITY APPLICATIONS

Loren R. Seidl
Section Head
Advanced System Security
& Network Technology
Hughes Aircraft Company
Microelectronic Systems Division

A distributed architecture design which interconnects terminals, microcomputers, and larger host computers must eventually face the question “who has access to this system?” Potential subscribers who do not have an appreciation for their need for security run the risk of a intermixed intercommunication network. There are typically additional responsibilities and security-related concerns about having controlled access. The introduction of operationally flexible microcomputers into systems also introduces a greater distribution of critical control tasks and greater access to sensitive communications. The classic solution would be to physically separate and thus electrically isolate inputs, outputs and the necessary computing hardware. Typically separate and dedicated connections would be used to securely pass data and control signals between the equipment. However, as a result of state of the art hardware design and software programming microcomputer technology is already waiting with programmable inputs/outputs, layered protocols, local area networks and powerful computing operations. Modularity, multi-tasking functions and conceptual system engineering have all contributed to the aforementioned integrated applications. Thus the need for an access control system which can be integrated into every connection of the system is inevitable.

Microcomputer tasks are usually programmed to be very efficient in accomplishing the assigned job. By designing a Separation Kernel with compartmentation and guarded privilege or discretionary data base access the assigned jobs can be performed without regard to security concerns. The resulting isolated data base could be containerized within the particular microprocessor RAM by using a variant addressing mechanism unique to each terminal. One example is the handling of both secure and clear digitized voice within a microprocessor based terminal. A Seperation Kernel directs the RAM access through a guard. This guard can be implemented either in hardware or software or a combination of both. However, verification of software separation is a task of varying difficulty. Whereas, hardware verification is relatively straightforward.

Maintaining the isolation of secure/clear (sometimes labeled Red/Black separation) over a common transmission medium requires careful design. In a distributed system, the definition of a connection-oriented TRANSMISSION SECURITY or TRANSEC based on initiator-connector protocol has many advantages. First it is relatively easy to appreciate from the users viewpoints. Secondly, if cryptographic devices are employed such as a Federal Data Encryption Standard (DES) implemented algorithm then mechanisms are definable for both initialization and synchronization of discretionary connections. One such application is illustrated in Figure 1 Security Protocol Design. Access to a connection can be granted or denied at the discretion of the initiator of the connection. The design protects against unauthorized intercept. The environment is defined to be nonmalicious i.e. no user is performing traffic analysis or imitative deception.

The DES design was chosen as the cover/decover pseudorandom algorithm implementation because of its validated correctness, its availability without T/SEC nomenclature and its 2^{56} or over 72 quadrillion possible encryption DES keys. The security protocol is built into a custom gate-array LSI and requires an initiator (m) of both secure and privacy connections to distribute its unique DES key and acknowledges by receiving back a message authentication code (MAC). The MAC is a function of the connectors' serial number SN (n), the received DES key (m), an initialization vector IV, and the DES algorithm itself. The MAC is then checked independently by the initiator. An audit count is kept by the initiator and when a connection is terminated the DES key of the connector is zeroized and the audit count adjusted. Connections are quickly made, authenticated and broken via the control bus. The physical separation of DES key transaction ports from distributed data bus ports prevent input/output failures which bypass security protocol.

The security protocol involves both operator and machine interactions. In order to ensure a secure connection the operator must not be allowed to bypass or ignore the operational steps. This implies that simple, well understood steps are needed and any default condition is safeguarded. The sequence of operational steps necessary to establish and maintain secure digital data on data bus is shown in the Security Operational Model flowcharts Figure 2 and Figure 3. This operational flow is dichotomized between initiator (Figure 2) and the corresponding connector action (Figure 3). Operator and machine actions are included in both cases. The load of a operational connectivity plan into each user terminal represents the initialization of the system. The supplied communication plan includes the user dependent connectivity matrix and the set of DES keys used to establish a user initiated connection. This communication plan is loaded via the control bus. A unique key is allocated for each of the permissable secure interconnections that can be supported by a particular user at any given time. Each key that is loaded into an individual terminal is checked for parity. If a parity error occurs, the operator is alerted and then takes corrective action by requesting a key reload.

At this point, the central initialization is complete and the initiator is ready to commence channel intialization. The initiating operator selects a communication channel and exercises the secure/clear option. The channel is checked against the connectivity matrix to verify that selected channel is a permissable interconnection. If a secure connection is required, the initiator sends a DES key via the control bus to all intended user connections. The recipient connector acknowledges by sending a message authentication code (MAC). This MAC is generated using the received key to encrypt the connector's serial numer. The initator then determines if the proper recipient connector has been contacted by deciphering the MAC and checking the received serial number against the allowable connection list. If a MAC error occurs, the operator is alerted and reestablishment of connection is attempted. At the time the DES key is received by the connector, a timeout/zeroize safeguard mechanism is activiated. This mechanism will zeroize the received connection key if the connector fails to respond to the initiation before a specified time has elapsed. Illuminated security indicators alert the operators at both the initiator and connector ends that secure communications are underway.

The initiator accounts for each connection key that is sent by using a key count register. This register is incremented each time the initiator sends a key to a particular connector. When a connection is broken, the initiator zerorizes the connection key at each user location and decrements the key count register. This audit trail mechanism prevents the erroneous storage of connection keys at any connector.

A distributed secure/clear architecture has been described that provides a limited, but useful form of connection oriented secure operation. The limitations are mainly in personal security controls and in distributing the DES key. Because the initiator itself controls discretionary access, the system must have some form of complete mediation over all of the terminals. A separate control bus has been provided which is used to initially load unique DES keys to each terminal. The central controller used to accomplish this task mediates individual secure access control. Through denial of a Des key the central controller inhibits any terminal from initiating a secure connection without proper authorization. If a terminal is removed from the system and a replacement terminal connected then the central controller must be notified.

Hughes MSD is actively pursuing the application of computer security architectures to systems containing microcomputers, terminals and intermixed data transmission mediums. Operational environments include both airborne and naval ship intercommunications. Distributed networks such as local area networks are rapidly contributing to the requirements of access control and secure connectivity. Standardization of secure algorithms is quite important to the further development of baselined generic systems. Enhancements can be developed for future multi-level security networks integrating both cryptographic and trusted software solutions.

ACKNOWLEDGEMENTS:

1. I would like to thank Rodney Threadgill, my staff associate at Hughes MSD for his valued technical assistance in the preparation of this paper.

REFERENCE:

1. Barnes, Derek H. "The Provision of Security For User Data on Packet Switched Networks," 1983 IEEE Symposium on Security and Privacy. Oakland, CA, pp. 4401-5.

2. Heider, George "Let Operating Systems Aid in Component Designs," Computer Design. Littleton, MA, Vol. 21, No. 9, Sept. 1982, pp. 151-9.

3. Klein, Melville H. "Computer Security," Signal. April 1983, pp. 11-9.

4. Schell, Roger R. and Cox, Lyle A. 'The Structure of a Security Kernel For a Z8000 Multiprocessor," 1981 IEEE Symposium and Security and Privacy. Oakland, CA pp. 124-9.

5. Stillman, R.B. and Defiore, C.R. "Computer Security and Networking Protocols," IEEE Transactions On Communications. Vol. com-28, No. 9, Sept. 1980, pp. 472-7.

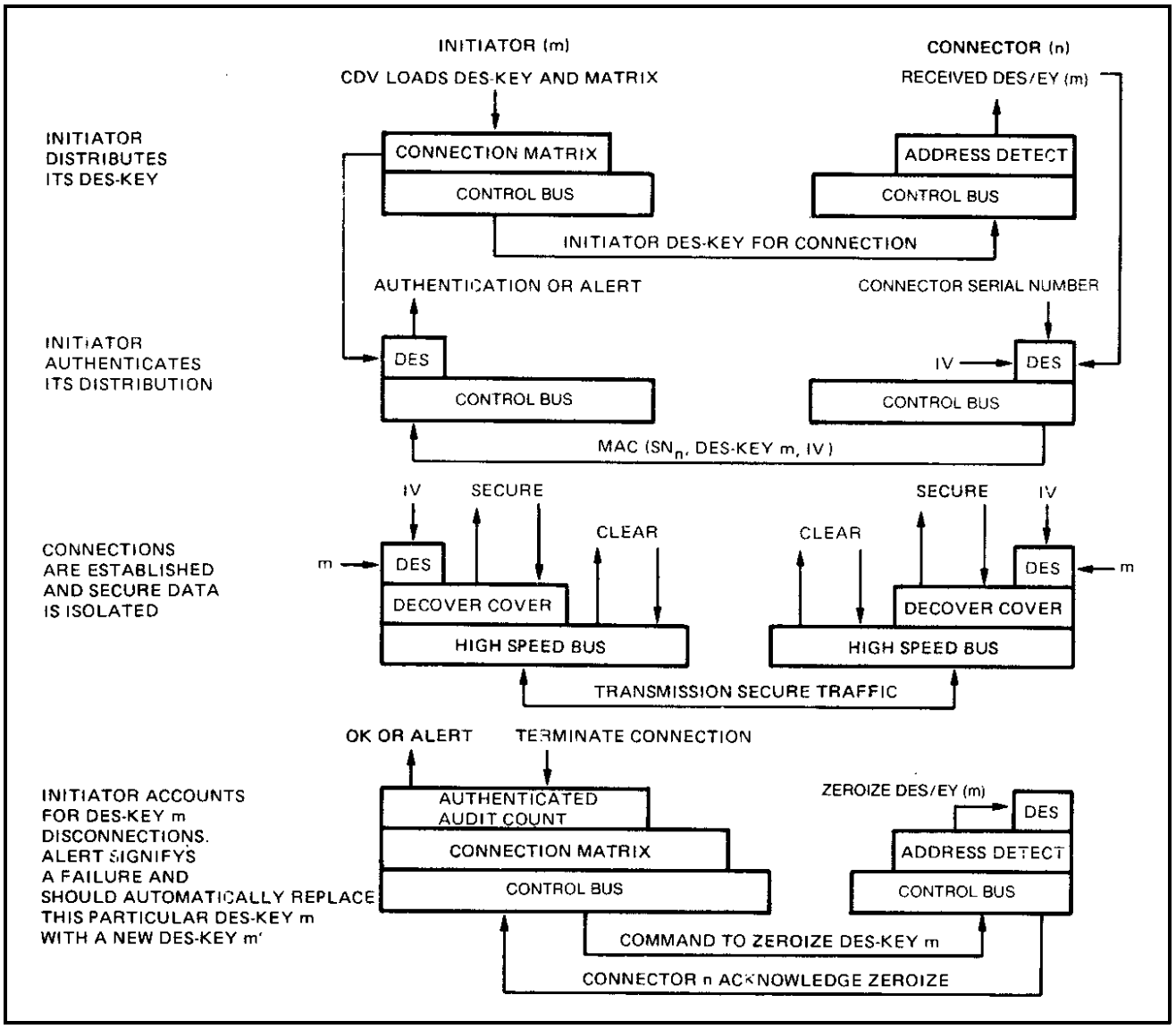

Figure 1 - Security Protocol Design

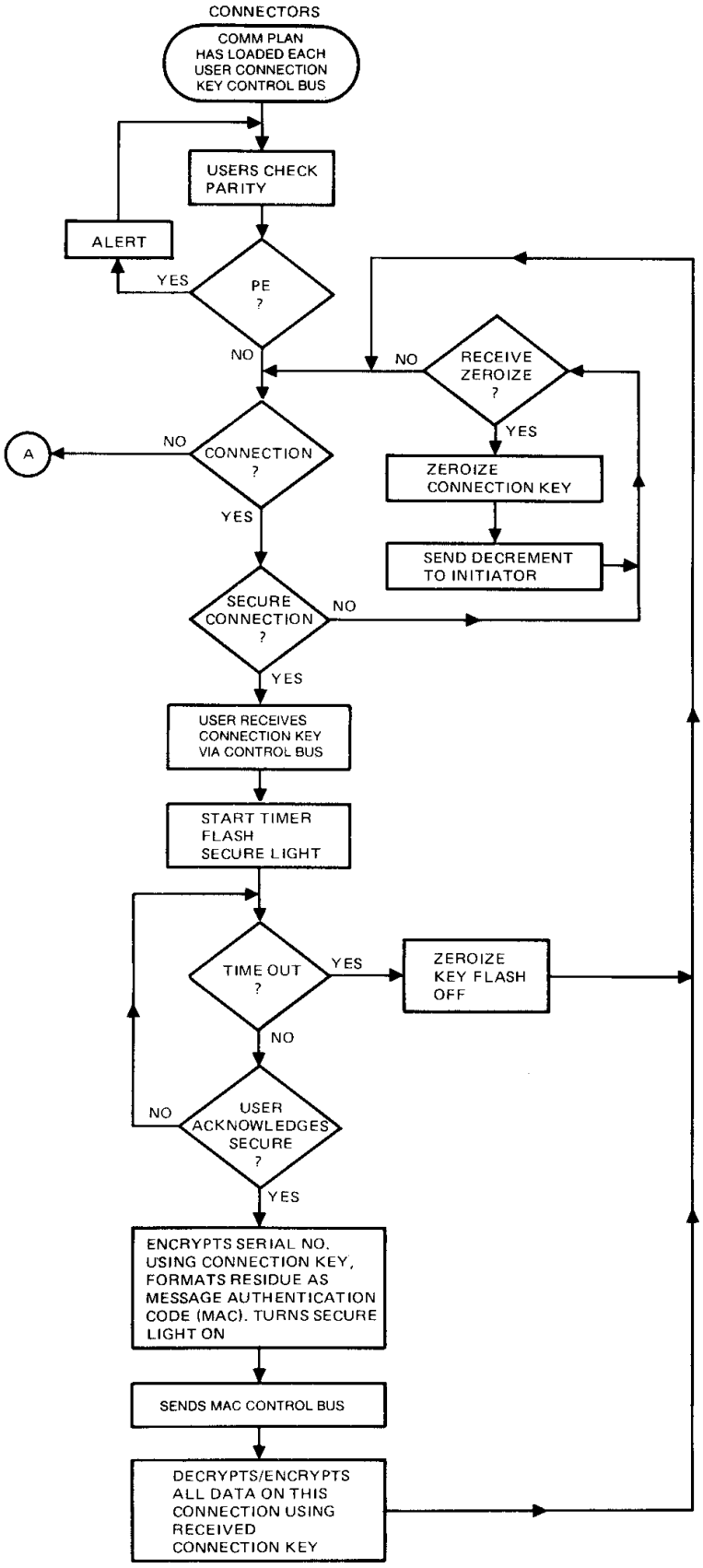

Figure 2 - Connection-Orientated Security Operational Model

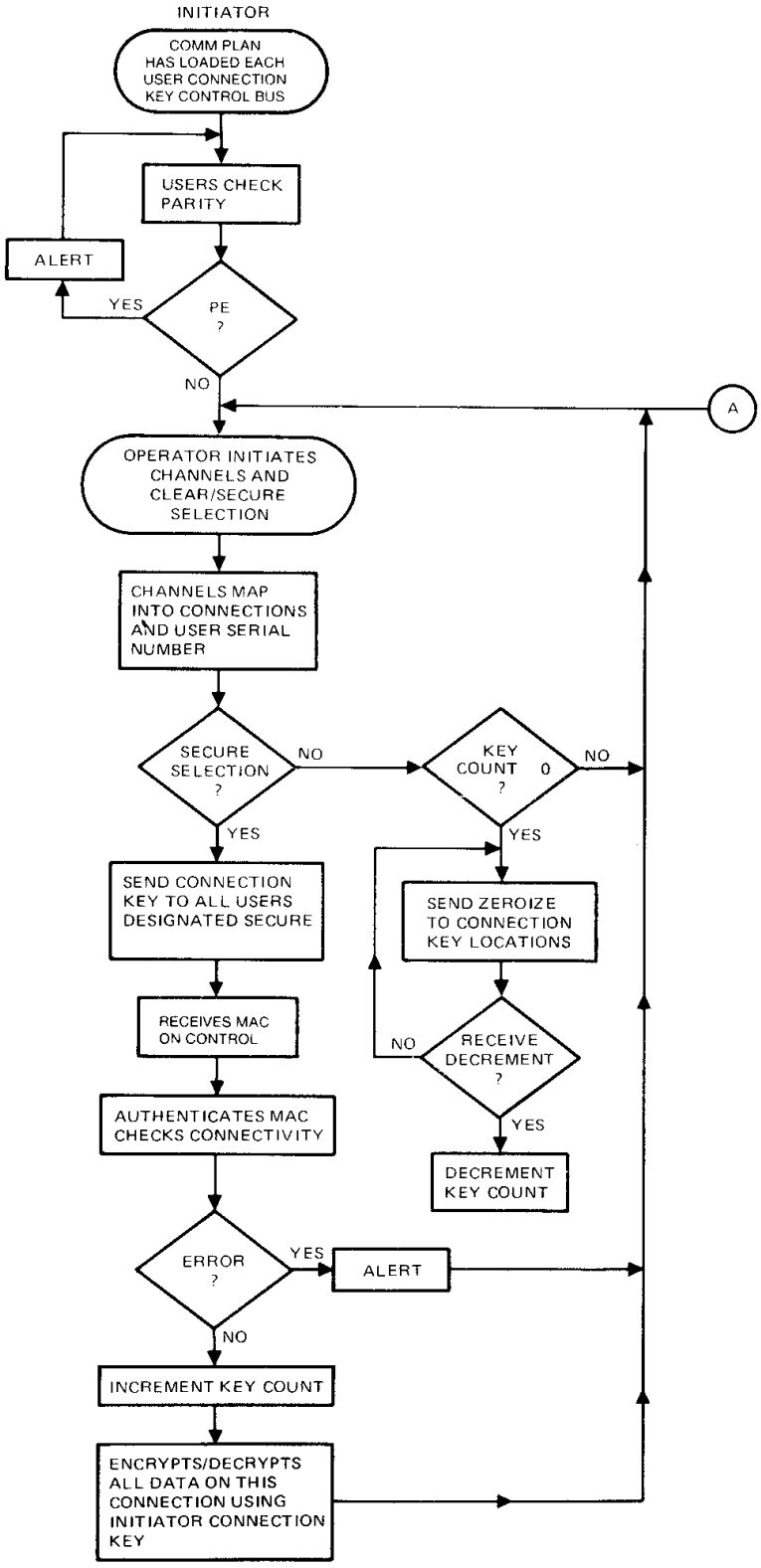

Figure 3 - Initiation Oriented Operational Model